

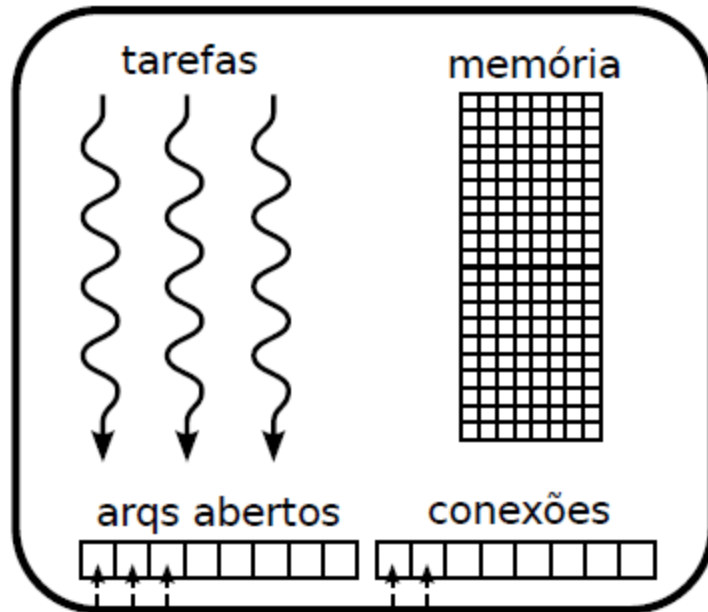
# Processos

# Introdução

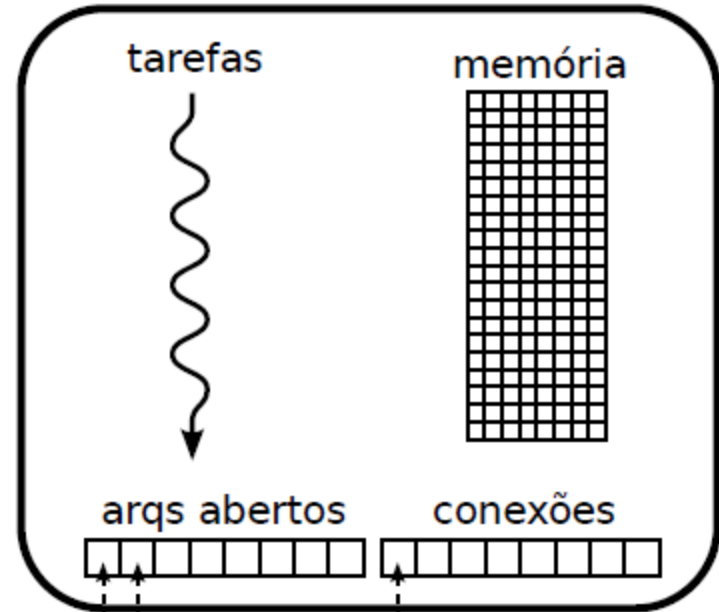
- Processador não conhece programas
  - Conhece instruções!
- SO deve gerenciar execução concorrente
  - Compartilhamento de UCP, E/S, memória principal, etc.
- Necessidade de algo que identifique a execução de um programa:
  - Tarefa, Job, Processo.

# Processo

Processo  $p_a$



Processo  $p_b$

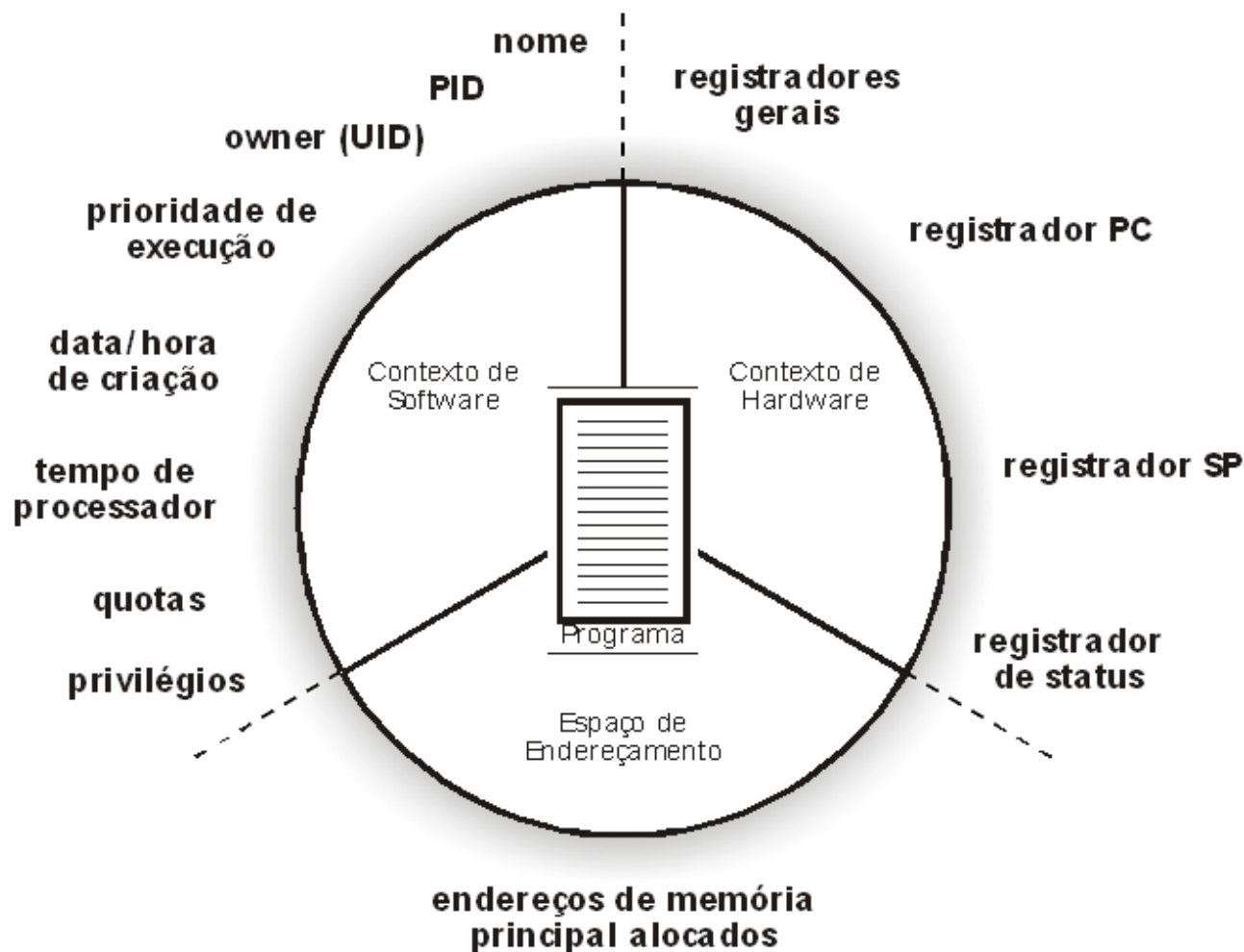


núcleo

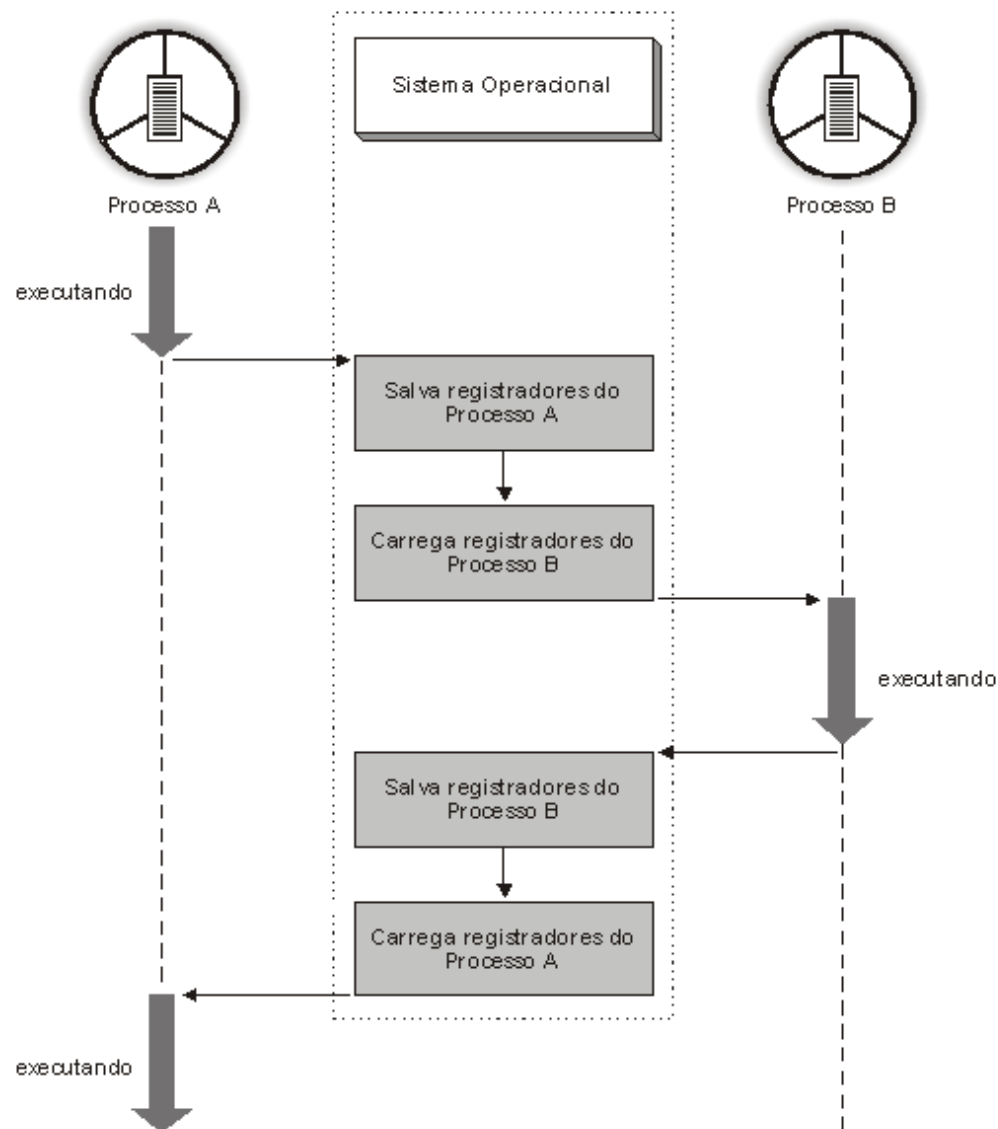
# Processo - Estrutura

- Processo
  - Programa em execução
  - Ambiente de execução desse programa.
- Troca de processos → Compartilhamento
  - Necessidade de salvar o contexto
- Formado por três partes:
  - Contexto de Hardware
  - Contexto de Software
  - Espaço de Endereçamento

# Processo – Estrutura - Contexto

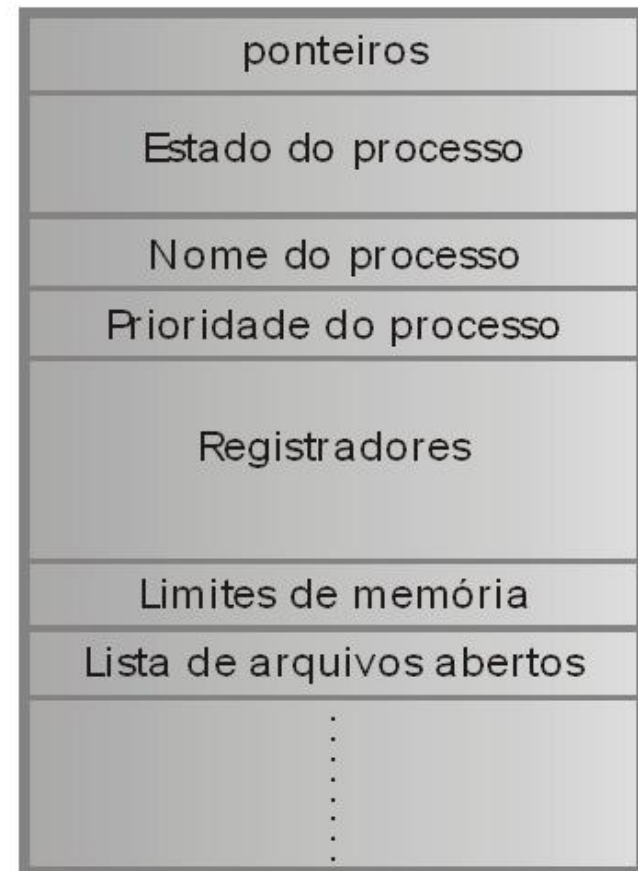


# Processo – Troca de Contexto



# Processo – Bloco de Controle

- Bloco de Controle do Processo (BCP)
  - Estrutura de dados
    - Informações sobre contextos
  - Implementação
    - Memória principal,
    - Área exclusiva do SO
  - System calls
    - Usadas na criação, alteração, eliminação, sincronismo, etc.



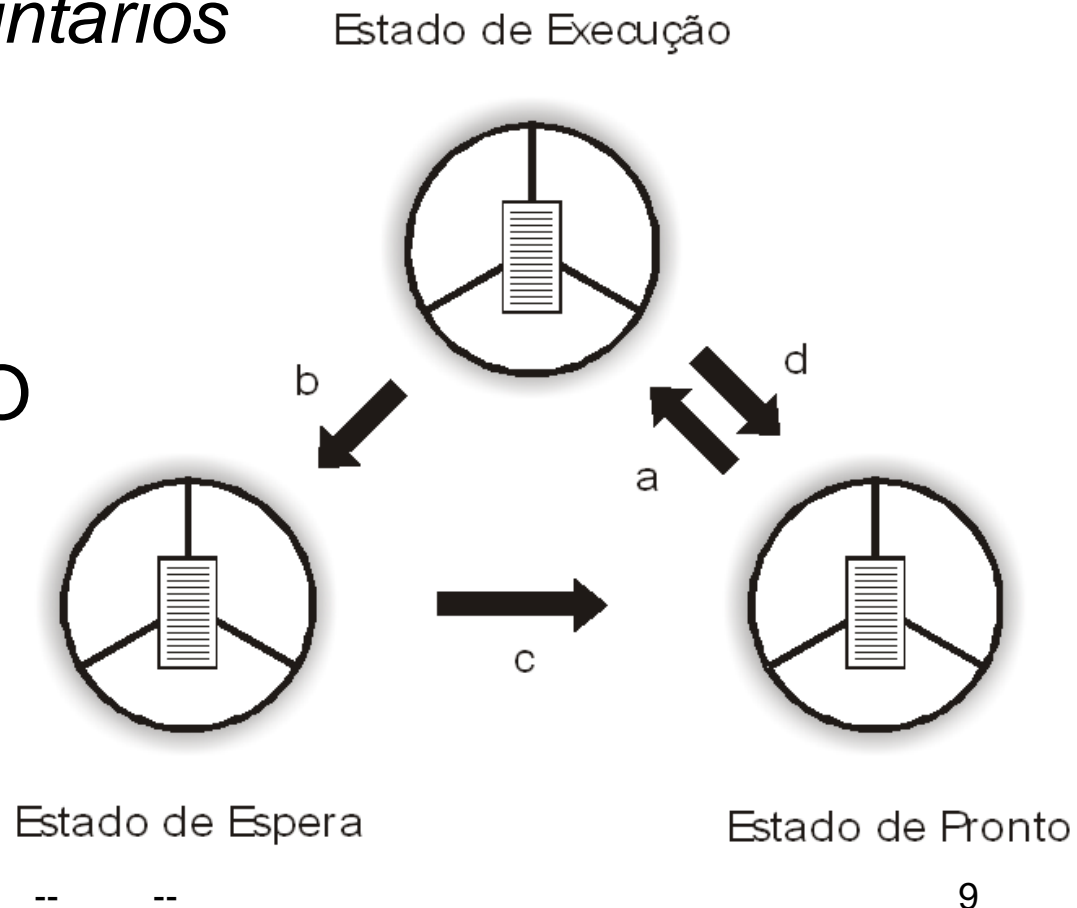
# Estados dos Processos

- Execução (*running*)
- Pronto (*ready*)
  - Lista de prontos
- Espera (*wait*)
  - Lista de espera



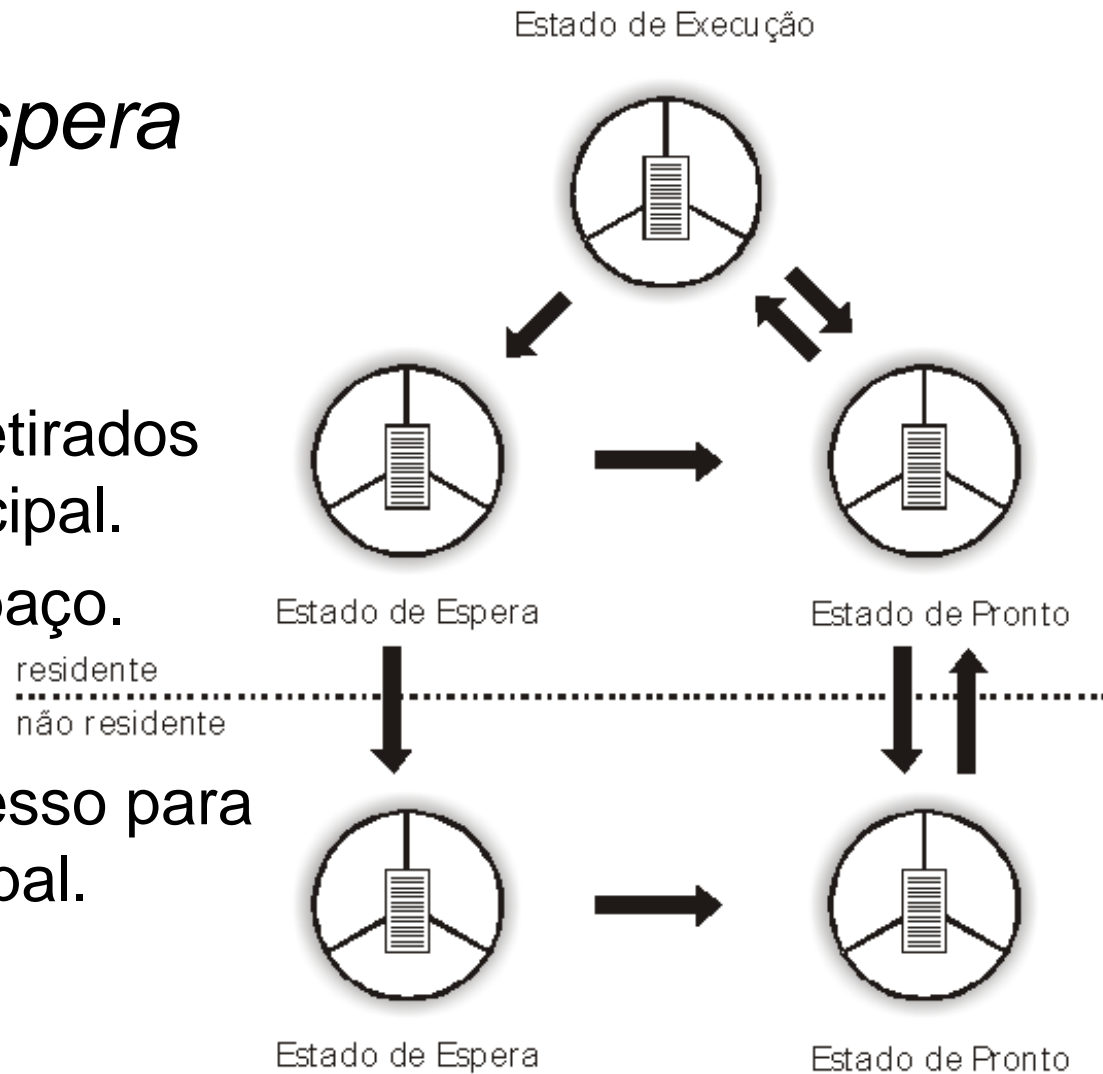
# Mudança de Estados

- Podem ocorrer:
  - Por eventos *voluntários* ou *involuntários*.
  - Políticas de escalonamento dependem do SO



# Mudança de Estados

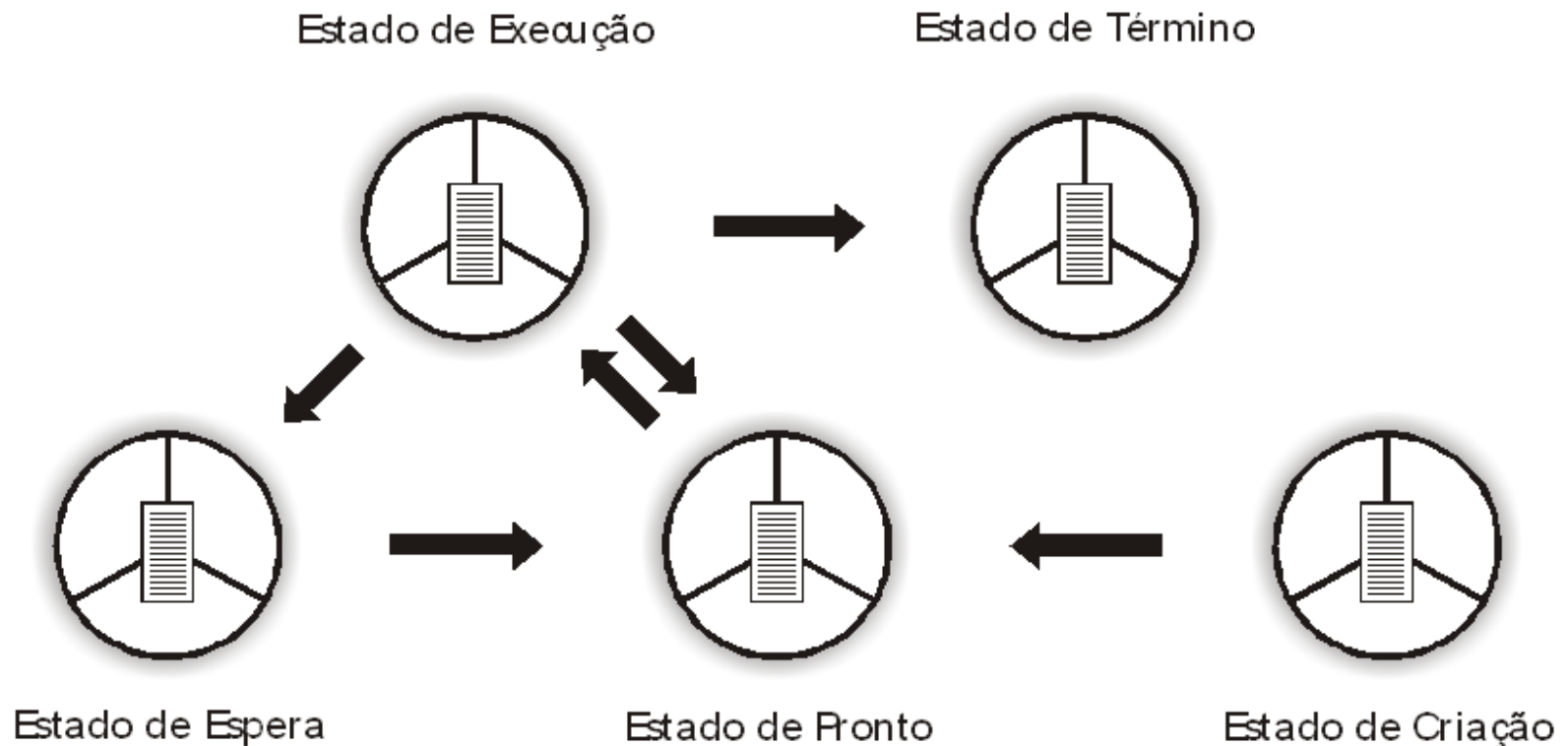
- Processos em *Espera* ou *Pronto*:
  - Swap out
    - Processos são retirados da memória principal.
    - Liberação de espaço.
  - Swap In
    - Retorno do processo para a memória principal.



# Criação e Eliminação de Processos

- Criação:
  - Sistema Operacional cria o PCB e aloca memória para uso do processo
  - A partir desse momento o processo já “existe” para o sistema Operacional
- Eliminação:
  - Todos os recursos alocados são desalocados
  - Exclusão do PCB.

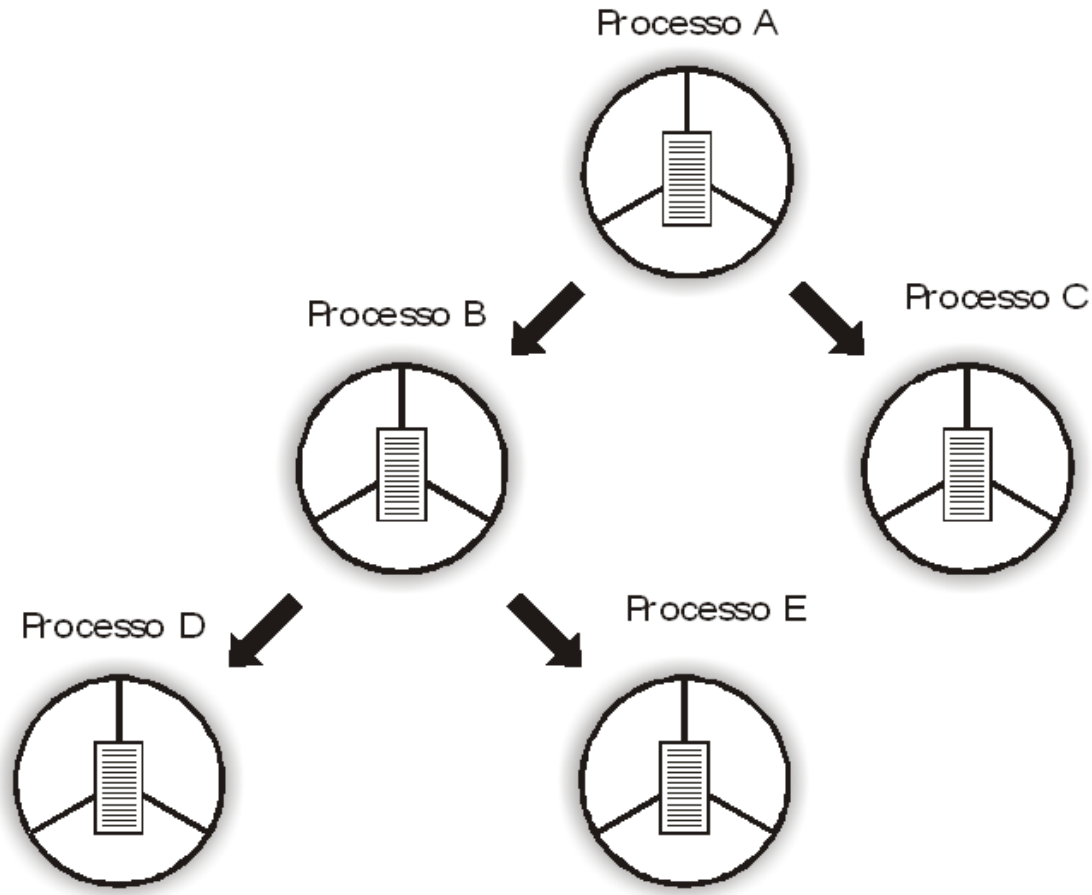
# Criação e Eliminação de Processos



# Processos Independentes e Subprocessos

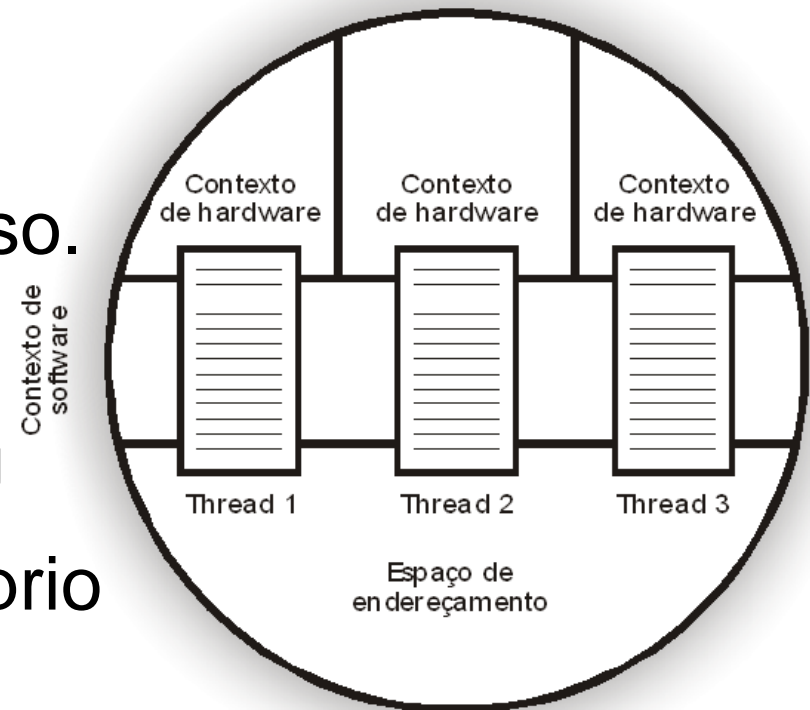
- Processos independentes
  - Sem vínculos com o criador.
  - Possui todos os contextos
- Subprocessos
  - Hierarquia de processos
  - Dependência entre processos
  - Compartilhamento de quotas
- Overhead na comunicação entre processos

# Processos Independentes e Subprocessos



# Threads

- Redução do overhead
- Um processo → várias linhas de execução.
- Contexto de Software e endereçamento do processo.
  - Maior desempenho na comunicação.
- Compartilhamento de CPU
- Contexto de hardware próprio



# Processos Foreground e Background

- Canais de comunicação (I/O)
- Foreground
  - Permite interação com o usuário
  - Canais: Dispositivos de E/S
- Background
  - Não há interação
  - Canais: arquivos, outros processos

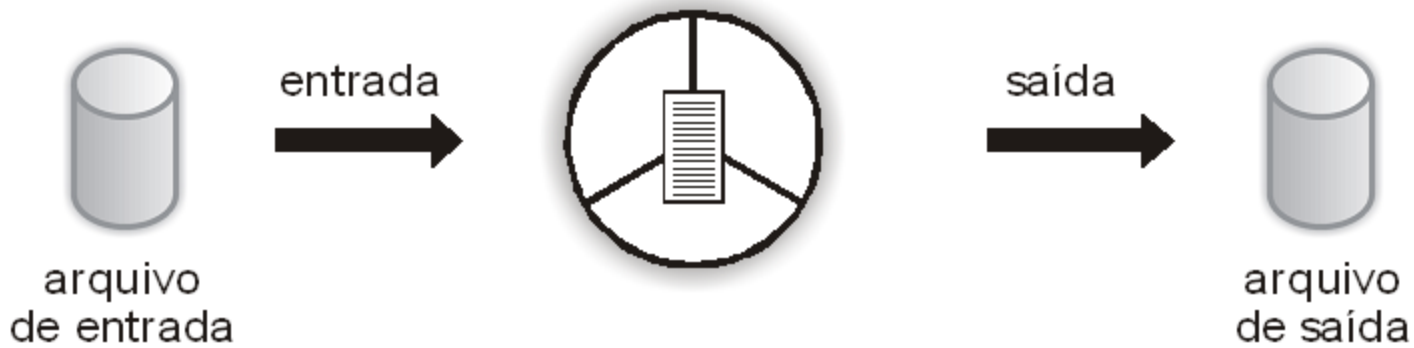


# Processos Foreground e Background

(a) Processo Foreground

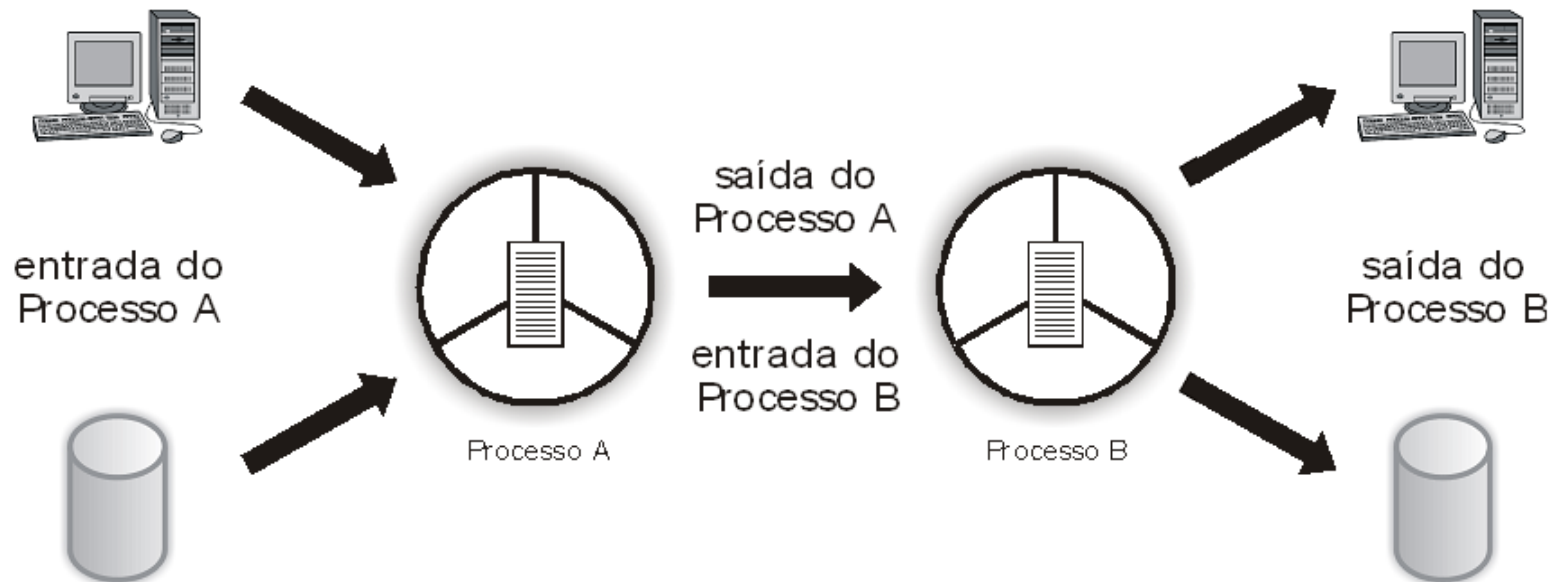


(b) Processo Background



# Pipe

- Saída de um processo ligado à entrada de outro.
- Muito usado nos interpretadores de comandos Unix.



# Processos do Sistema Operacional

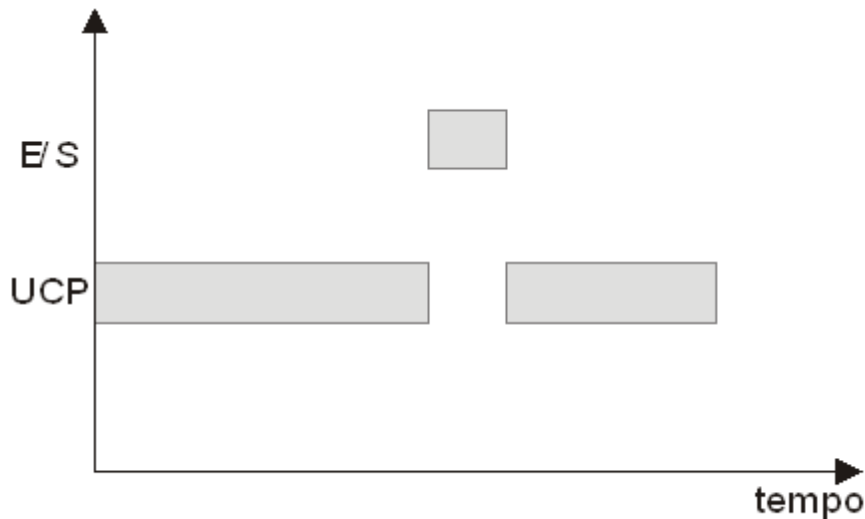
- Não só aplicações usam processos
- Arquitetura Microkernel
  - Núcleo estável
  - Flexibilidade de ativar/desativar recursos
    - Ex: Serviços de Rede
    - Ex: Auditoria e segurança
    - Ex: Gerência de impressão

# Processos CPU/IO - Bound

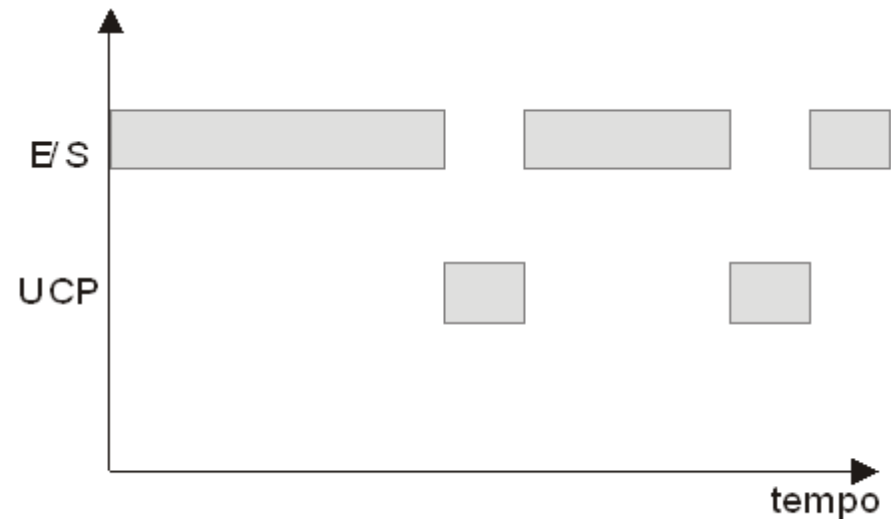
- Classificação de processos de acordo com sua utilização.
- CPU-Bound
  - Estado de **execução**
  - Aplicações que efetuam muitos cálculos (científicas, processamento de imagens, etc.)
- IO-Bound
  - Estado de **espera**
  - Aplicações comerciais (muitas leituras e gravações)

# Processos CPU/IO - Bound

- Comparação entre processos CPU-Bound e processos IO-Bound.



(a) CPU-bound



(b) I/O-bound

# Sinais

- Sinais
  - Notificação de processos
  - Usados com temporizadores para execução periódica
  - Notificação no PCB
    - Alteração de bits
    - Verificação apenas na execução.
  - Tratamento semelhante ao de interrupções/exceções
    - Ex.: Ctrl+C, Término do processo

